REMARKS

Claims 13-27 are pending. Claims 13, 17 and 21 were amended to recite inherent features of the present invention.

No new matter was added. The claim phrases added to the independent claims are fully supported by at least page 3, lines 15-22, and page 3, line 29 through page 4, line 2 of the specification.

Examiner Interview

Applicants wish to thank Examiner Siddiqi for extending the courtesy of a telephone interview in respect to this application on December 10, 2008 with Applicants' undersigned representative. During the interview, a previously filed "Response After Final Under 37 CFR 1.116" (hereafter, "the Response After Final") was discussed. The following items were discussed:

1. Applicants reviewed the arguments directed to the specification objection set forth on pages 3-4 of the Response After Final, especially pointing out that the claim format of claims 17-20 is explicitly permitted by the USPTO and should be treated as a statutory product claim, as discussed in MPEP 2106.01. Applicants further pointed out that this position is further supported by *Ex parte Bo Li*, Appeal 2008-1213 (USPTO BPAI 2008, November 6, 2008) (copy enclosed as an Appendix to this paper). See, especially, page 9 of this opinion. In the Advisory Action dated December 11, 2008, the Examiner stated that this specification objection is withdrawn.

2. The Examiner stated that claims 17-20 will be reevaluated to determine whether they meet the requirements of 35 U.S.C. § 101. Applicants pointed out that the arguments related to the specification objection similarly show that claims 17-20 meet the requirements of 35 U.S.C. § 101.

3. Applicants reviewed the arguments directed to the prior art rejection set forth on pages 5-8 of the Response After Final, especially the argument that the URL referred to on column 8, line 54 of Loomans that was alleged to disclose the claimed URI of a web page is not a URI or a

URL of a web page, but instead is merely a URL of the kernel that is downloaded into the browser used in Loomans, and that a kernel is not a web page. The Examiner explained that the kernel is used by a browser in forming a web page, but Applicants' representative pointed out that this does not read on the explicitly recited claim language which requires the code to include a URI of a web page. The Examiner further stated that the mere presence of a URI or URL in the script code of Loomans is sufficient to meet the claim language, but Applicants' representative pointed out that this is a clearly erroneous position because of the explicitly recited claim language which requires the code to include a URI of a web page. The Examiner agreed to further reconsider his position.

4. Applicants' representative argued that not only does Loomans fail to disclose code that includes a URI of a web page, but that the URI of the web page is used for a specific authorization purpose which is completely absent from Loomans. The Examiner stated that the authorization purpose was not set forth in the claims. Applicants' representative responded that it is not necessary to recite the authorization purpose because the claims already explicitly recite that the code includes a URI of a web page and this feature is absent from Loomans. Notwithstanding this fact, the Examiner stated that it would be helpful to recite this function so as to further distinguish the claims from Loomans. To advance prosecution of this application, the claims have been amended to recite the authorization function. Since no such function is even remotely disclosed or suggested by the URI in Loomans, Applicants believe that the amended claims are now even more clearly patentable over Loomans.

5. The Examiner stated that the claims appear to merely recite an editor. In response, Applicants do not concur in this narrow characterization of the claims. Furthermore, even if the claims are readable on an editor, the claims are still believed to be patentable over any editing process that occurs in Looman for the reasons discussed above. Stated another way, if the claims are to be rejected over an editor, then the editor must perform or suggest the explicitly recited steps of the claims, including step (b)(i). Loomans does not perform or suggest at least this step.

The arguments presented in the Response After Final are repeated below with minor changes to reflect the amended claim language.

Specification Objection

The specification is believed to be moot in view of the Examiner's withdrawal of this objection in the Advisory Action.

Rejection under 35 U.S.C. § 102

All pending claims were rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by Loomans. Applicants respectfully traverse this rejection.

1. Loomans

Loomans discloses a process for deploying a generic application engine in a browser program executing on a client platform. Referring to Fig. 6, the process operates as follows:
1. A user loads the application engine shell page into a browser using an associated URL (step 602).
2. URL references within the frameset initiate loading of the kernel into the browser (step 604). The kernel is formed of code that is generally script embedded in HTML, such as Javascript or VB script. The Javascript (code) provides basic thread management functions such as event handling routines, blocking functions, and structures (such as frames) within which application components such as data and additional code may be loaded later, if required (column 5, lines 22-27).
3. The kernel, in turn, requests a minimum required subset of application engine components to load into the browser (step 606). These are the components that will be required for the user to proceed at the first page.
4. Once loaded, the application engine kernel runs any required initialization routines and initiates loading of any additional application engine components that are not required at starttime (step 608).
5. Concurrently with the bootstrapping[1] process in steps 604-608, the application engine shell

[1] Bootstrapping refers to techniques that allow a simple system to activate a more complicated system.

page loads initial User Interface (UI) components and data components associated with the initial sub-application to be deployed (step 610).

6. Subsequently, the user provides requested inputs to the displayed UI components (step 612).

7. The application engine processes the user provided inputs in the particularized context of the sub-application's data and any API extension code (step 614).

8. The corresponding results are generated from the data and displayed in additional HTML in response to the user-supplied inputs (step 616).

9. A determination is made whether or not a new sub-application is to be processed (step 618).

10. If there are no additional sub-applications to process, the application simply waits for the next user input in the current sub-application. However, if there are additional sub applications to process, control is passed to step 620 where the additional components are loaded and control is then passed back to step 612.

2. Patentable differences between Loomans and claimed invention

Except for the fact that Loomans constructs a web page that includes script, Loomans has nothing else whatsoever to do with the claimed invention. (Applicants are not claiming to have invented constructing a web page that includes script.) The following table provides a snapshot view of the deficiencies in Loomans:

Claim language	Portions of Loomans highlighted by Examiner	Applicants' rebuttal comments
13. A method of constructing a web page that allows for receipt of digital assets, the method comprising:	Examiner asserts that Loomans provides this function.	No digital assets[2] are received in Loomans. In fact, digital assets are not discussed anywhere in Loomans.

[2] A digital asset is any form of content and/or media that have been formatted into a binary source which include the right to use it. A digital file without the right to use it is not an asset. Digital assets are categorized in three major groups which may be defined as textual content (digital assets), images (media assets) and multimedia (media assets). Wikipedia definition of digital asset at: http://en.wikipedia.org/wiki/Digital_asset, attached hereto as an Appendix)

(a) constructing a web page; and	Step 616 of Figure 6; column 9, lines 3-6	
(b) inserting into the web page script	embedded kernel 222 in Figure 2; column 6, lines 54-61	The Examiner is presumably referring to the fact that the kernel is formed of code that is generally script embedded in HTML, such as Javascript or VB script.
associated with at least one digital asset that is desired to be part of a fully rendered web page,	embedded kernel 222 in Figure 2; column 6, lines 54-61	As discussed above, the kernel and its associated script performs the following functions: 3. The kernel, in turn, requests a minimum required subset of application engine components to load into the browser (step 606). These are the components that will be required for the user to proceed at the first page. 4. Once loaded, the application engine kernel runs any required initialization routines and initiates loading of any additional application engine components that are not required at starttime (step 608). The script in Loomans is not associated with at least one digital asset that is desired to be part of a fully rendered web page. Stated simply, the script in Loomans serves a completely different purpose than the script in the claimed invention.
the inserted script including code to request the content of the digital asset from a remote site when the code is executed by a browser,	client computer 202 in Figure 2; column 6, line 36 through column 7, line 2	None of the functions performed by the kernel and its associated script request the content of [a] digital asset from a remote site.
the code including: (i) a uniform resource identifier (URI) of the	URL referred to on column 8, line 54	The "URL references" on column 8, line 54 is not a URI of a web page. Instead, it a URL (location) of the kernel that is

web page for use by the remote site in authenticating whether the URI is authorized to receive the content of the digital asset, and		downloaded into the browser. A kernel is not a web page. There is no disclosure anywhere in Loomans of the concept of script code that includes a URI (or its equivalent) of a web page for use by the remote site in authenticating whether the URI is authorized to receive the content of the digital asset. This feature allows the remote site to verify that the request for the content of a digital asset has been made from a web page that has the appropriate permission to receive the content of the digital asset. In one preferred embodiment, the remote site compares the URI to an authorized list of URI's and only returns the content of the digital asset if the URI is authorized to receive it. Loomans has no authorization/permission step, so there is no reason to even add the concept of script code that includes a URI (or its equivalent) of a web page for use by the remote site in authenticating whether the URI is authorized to receive the content of the digital asset to Loomans.
(ii) a unique identifier of the content of the digital asset.	elements of Figure 6 (no specific elements were highlighted); column 8, lines 49-67	Loomans has nothing to do with "content of [a] digital asset," and thus the Figure 6 process described above inherently does not insert a unique identifier of content into the script code.

3. Patentability of independent claims 13, 17 and 21 over Loomans

Claim 13 reads as follows (underlining added for emphasis):

> A method of constructing a web page that allows for syndication of digital assets, the method comprising:
> (a) constructing a web page; and
> (b) inserting into the web page script associated with at least one digital asset that is desired to be part of a fully rendered web page, the inserted script including code to request the content of the digital asset from a remote site when the code is executed by a browser, the code including:
> (i) a uniform resource identifier (URI) of the web page for use by the remote site in authenticating whether the URI is authorized to receive the content of the digital asset, and
> (ii) a unique identifier of the content of the digital asset.

As discussed above in the table, Loomans does not disclose any of the above-highlighted features. Nor does Loomans have any disclosure that suggests such features. Accordingly, claim 13 is believed to be patentable over Loomans.

Claims 17 and 21 are also believed to be patentable over Loomans for the same reasons as applied to claim 13.

4. Examiner comments in Advisory Action

The Advisory Action merely refers to the same sections of Loomans as referred to in the Final Rejection, and thus presents no new position regarding the applicability of Loomans to the claims. Accordingly, the comments above are believed to be fully responsive to the Advisory Action statements regarding Loomans.

5. Patentability of dependent claims

The dependent claims are believed to be patentable over the applied references for at least the reason that they are dependent upon allowable base claims and because they recite additional patentable elements and steps.

6. Patentability of claims over prior art in outstanding EPO Office Action

Although not applied against the pending claims, such claims are believed to be patentable over the prior art references applied in the outstanding EPO Office Action submitted with the Supplemental Information Disclosure Statement filed on September 11, 2008. The following brief comments are provided regarding such references.

a. WO 00/20945 (Yee). Yee does not insert into a web page script associated with at least one digital asset. Nor does the use of COM objects in Yee have anything to do with this feature. Furthermore, since there is no script in Yee, there is no disclosure in Yee of the concept of script code that includes a URI (or its equivalent) of a web page. As discussed above, this feature allows the remote site to verify that the request for the content of a digital asset has been made from a web page that has the appropriate permission to receive the content of the digital asset.

b. U.S. Patent No. 5,999,941 (Andersen). Andersen also does not insert into a web page script associated with at least one digital asset. Andersen discloses the use of an applet (which is not script) within a browser of a personal computer 103. The applet prepares a URL to address an active server page (ASP) on a server computer 208. The "script" in Andersen is stored in the active server page (ASP) of the server computer 208. Also, the script in Andersen does not have code that includes a URI (or its equivalent) of a web page being constructed. Anderson has no authorization/permission step, so there is no reason to even add the concept of script code that includes a URI (or its equivalent) of a web page to Andersen.

In sum, neither of these references are relevant to the claimed invention.

Conclusion

Insofar as the Examiner's rejections were fully addressed, the instant application is in condition for allowance. Issuance of a Notice of Allowability of all pending claims is therefore earnestly solicited.

Respectfully submitted,

RICHARD D. MARTIN et al.

December 15, 2008 By: Clark Jablon
(Date)

CLARK A. JABLON
Registration No. 35,039
PANITCH SCHWARZE BELISARIO & NADEL LLP
One Commerce Square
2005 Market Street - Suite 2200
Philadelphia, PA 19103
Telephone: (215) 965-1330
Direct Dial: (215) 965-1293
Facsimile: (215) 965-1331

Enclosure: Appendix (12 pages)